Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2018 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

6-26-2018

Developers' Core Practice in Technology Startups: Toward a Grounded Theory

Ma. Beth Concepcion
De La Salle University, ma_beth_concepcion@dlsu.edu.ph

Raymund Sison
De La Salle University, raymund.sison@delasalle.ph

Developers' Core Practice in Technology Startups: Toward a Grounded Theory

Completed Research Paper

Ma. Beth S. Concepcion
De La Salle University – Manila
Taft Avenue, Manila, Philippines
ma_beth_concepcion@dlsu.edu.ph

Raymund C. Sison
De La Salle University-STC
Laguna, Philippines
raymund.sison@delasalle.edu.ph

Abstract

Technology startups are globally recognized as contributors to innovation and entrepreneurship. However, this emerging industry faced a lot of challenges and uncertainties in the search for a scalable business model. Considering the importance of people as innovative drivers in the development of startups, this study uncovers what constitute to the experiences of software developers using a Grounded Theory Methodology. Findings revealed that acquiring skills in doing multiple roles is their main concern and the core category that resolves their concern is the cyclical basic social process of Roling. Roling involves two stages: Instalearning and Perfortraying. Instalearning is employed through the following techniques: self-learning, collaborating, and networking. Perfortraying includes the following practices: Dutying, Spanning, Substituting, Sharing, Swapping, Grabbing, and Hopping. This study is hoped to contribute to the development of processes, structures and policies that highlights the role of developers in technology startups among other stakeholders.

Keywords: Software developer, Technology startups, Grounded Theory Methodology

Introduction

Startups have been widely recognized as engines that promote innovation, steer economic development, and create job opportunities (Wiens and Jackson 2014, OECD 2013). Its global diffusion can be seen in the rapid burst of high-growth technology products and companies (Herrmann et al. 2015). Even though this phenomenon has growth potentials in the Philippine context (Hose 2013), still many are striving to survive because of uncertainties brought by socio-economic, environmental and technological factors.

The challenges faced by technology startups are far different from established companies and organizations. Startups are characterized by flat structure, small size, limited resources and often exposed to pressure, uncertainties, risks and rapid changes in technologies (Paternoster 2014). Startups' speed is faster than a large company (Blank and Dorf 2012) and their focus is finding the business model, which is relatively existent in established organizations (Ries 2010). Startups do not have a well defined-process or procedure, which is the antithesis of large enterprises (Blank 2007). In addition, startups are not smaller versions of large companies with specific master plans because they often encounter ultimate success and quick failure (Blank 2013). In some cases, they share similar characteristics with small companies but a combination of these characteristics makes startups unique and therefore affects the software development activities (Paternoster et al. 2014) and also the practices of workers.

The systematic mapping study conducted by Paternoster et al. (2014) highlights that there is much to explore in the context of startups because the existing body of knowledge is still limited to a few high quality studies. In their research agenda, Unterkalmsteiner et al. (2016) identified research gaps in the area, highlighting the human aspect as one of the interesting areas of study. Specifically, they direct future researchers to study actors in software startups particularly on how they work.

The aforementioned studies and observations suggest the need to explore the technology startups by looking at the other angle- from the experiences and insights of software developers. The paucity of studies focusing on software developers' experiences in startups and the absence of it in the Philippines motivate the researcher to conduct an in-depth study of their experiences. Using a Grounded Theory Methodology (GTM), this preliminary investigation aims to provide a substantive evidence of what is going on in their fields, their major concern, the factors that greatly affect them, and the processes that aid them in resolving their concern.

The succeeding sections of this paper present the research design to include the research settings and subjects, modes of data collection, and how a grounded theory methodology is being employed in this study. This will be followed by the discussion of results and theoretical integration with existing literature. Lastly, this paper presents conclusion and recommendation.

Research Design

This research aimed to account what is happening in the substantive area: software developers in technology startups in the Philippines. Since this study used the Glaserian Grounded Theory Method (GTM), the researchers are encouraged not to enter the field with specific narrow questions. Instead, a broad problem definition in a very general form is utilized (Adolph et al. 2012). Thus, this study was guided by this question: *What is the main concern of software developers in technology startups in the Philippines and how did they resolve this concern?*.

The goal of this research is to discover the main concern and the core category that reflects the process on how they resolve their main concern. The core category according to Glaser (1998) is "the pattern of behavior which is most related to all other categories and their properties in the theory which explains how participants resolve their main concern".

Research Settings and Subjects

This study focused on the substantive area of software developers in technology startups in the Philippines. While the government, private sectors, academe, and startup communities are working collaboratively in order to capacitate this emerging industry, still there is a loose focus on the concerns and issues of people working in this sector. In this case, we deemed to explore the first hand experiences of workers specifically those involved in the development roles in technology startups.

We were able to study the substantive area coming the following cities across the country: Metro Manila, Iloilo City and Cebu City. A total of 24 respondents coming from 13 technology startup companies participated in the study (see Table 1). In looking for respondents, we established connections with startup societies and developers' communities and asked them for referral. We also joined startups' events and developers' meetups and took the opportunity to network with possible participants. In addition, we went to various incubation hubs and working spaces across the country to look for respondents.

The selection of respondents for this study was based on the initial criteria: (1) He or she must be a software developer, and (2) He/she must have at least 1 year of working experience. However, as the research progressed, we saw the need to include developers who have founder or manager roles and those who have less than 1 year of experience. This decision was made based on theoretical sampling, which dictates on what data to sample next.

Data Collection

This study explored the substantive area through in-depth and detailed data collection. We conducted in-depth interviews with developers working in the context of technology startups. Interviews were done in their respective offices, co-working spaces, incubation facilities, or in the coffee shops. Interviews were semi-structured and open-ended. Conversations with them vary from 45 minutes to 105 minutes. These were done face-to-face to allow the key informants to raise concerns out of their own will and for us to capture their responses clearly. After seeking their consent, we took field notes as a basis of coding and audio-recorded the interviews as a source of verbatim indicators. Interviews were transcribed manually for further analysis. Most of the interviews were conducted using an English language. However, some used their local languages or dialects; these were further translated to English.

Data were also obtained through direct observations of their practices and t working environment. We spent at approximately 3 hours for on-site observation and interview per startup company. In addition, we also joined their events, meetups and conferences to gain additional insights of their activities.

Table 1. Sample Respondents

Startup Company	Location	Assigned Roles of Respondents	No. of Respondents/ Company	Age of Respondents (Average)
S1	Iloilo City	P1=D; P14=PM&D	2	24.5
S2	Iloilo City	P2=F&D; P3=F&D	2	24.5
S3	Iloilo City	P4= CTO&D; P15=D	2	31.5
S4	Metro Manila	P5= F&D; P6=D; P7=D	3	29
S5	Metro Manila	P8=F&D; P9=D	2	27.5
S6	Metro Manila	P10= F&D; P11=D; P12=D	3	24.67
S7	Metro Manila	P13=F&D	1	22
S8	Iloilo City	P16=D	1	26
S9	Cebu City	P17=F&D; P18=D	2	25.5
S10	Cebu City	P19=F&D	1	34
S11	Cebu City	P20= F&D	1	35
S12	Cebu City	P21= F&D; P22=D,	2	26
S13	Cebu City	P23=D; P24=D	2	23
Sum: D= 12 ; F&D=10; CTO&D=1; PM&D=1 Manila=9P; Iloilo=7P; Cebu=8P Total No. of Respondents			24	27.17 (Average)
Legend: D = Developer; F&D = Founder and Developer CTO&D = Chief Technology Officer and Developer PM&D = Project Manager and Developer				

The Glaserian Grounded Theory Methodology

This study is inspired by Glaserian approach of Grounded Theory (Glaser and Strauss 1967), an inductive approach in generating a substantive theory that explains the meaningful accounts of what is going on in the field of software developers in the context of technology startups. This method is particularly helpful for researches to predict and explain actors' behavior (Goulding 2002), understand their own working environment (Martin and Turner 1986), and construct meaning out of their inter-subjective experiences (Sudabby 2006). This study chose to employ this method because of rigor and relevance. In particular, GTM uses a stringent and simultaneous data collection and analytic procedure in the interpretation and presentation of findings (Fernandez et al. 2002), which is

manifested by the unique methodological elements called the constant comparative method and theoretical sampling (Suddaby 2006).

In studying the substantive area, we had a careful consideration of the GTM tradition that no predefined constructs or guiding theories before entering the field. Thus, pre-literature review was avoided to ensure that the emergence of categories would not be contaminated and constrained by the prior concepts (Glaser 1992).

We then proceed to the next step, identification of potential core category. This step includes data collection, coding and memoing, which are simultaneously and iteratively done using the constant comparative method. The constant comparative method enables the generation of theory through a systematic and explicit coding and analytic procedures (Glaser 2004). We utilized the open coding technique to analyze the data, extract a set of categories and their properties, produce a higher level of abstraction, and induce patterns in the data. We coded everything while going through the data and then we employed a constant comparison. The purpose of open coding is to identify the concepts, which repeatedly appear in the data. In the process of coding, we constantly refit the concepts and categories to the data as the research proceeds. The sample coding process is shown in Table 2.

Table 2. Sample coding process

Incidents/	Codes	Memos
I have a good problem-solving skills but I didn't have an experience in business.	Problem-solving skills (strength) Business (weakness) Lack of skills	For P2, the challenging part of doing his role(s) is the lack of skills in business. Learning is a technique P2 used to address the challenge brought by the lack of skills
So working here needs me to learn skills in connection with business, processing business permits, marketing etc.	Learning	A specific technique of learning (gaining insights from others) is through Networking.
So I need to meet certain people and talk with them to gain insights.	Networking	
I am Founder and developer.	Multiple roles	

After identifying potential core category, we ceased open coding and proceeded to selective coding. We employed the selective coding in order to delimit coding only to those variables that relate to the core category. Thus, theoretical sampling was more selective and theoretical memos become more focused to the aspect of the core category. These activities were done in iteration until it reaches theoretical saturation. Theoretical saturation occurs when "no additional data are being found whereby the analyst can develop properties of the category" (Glaser and Strauss 1967, p.61).

Results and Discussions

After abstracting the codes and identifying its properties, we were able to organize these concepts to form a higher-level category. By applying the grounded theory method in the substantive area, the emerging codes, categories, and properties were further analyzed through constant comparison, conceptual memoing and theoretical coding. Thus, the preliminary results revealed that acquiring skills in doing multiple roles indicates the main concern of software developers in technology startups and the core category that resolves this main concern is the cyclical basic social process of Roling.

The Main Concern in the Substantive Area

This study considers that acquiring skills in doing multiple roles is the main concern of developers in technology startups. Since technology startups size is small, developers tend to play various roles to include both technical and non-technical. They are not only bounded to perform based on their assigned roles but also assumed other roles. In many cases, they often take additional roles that are not typical for their job titles.

Performing various roles posed pressing concern to developers because their preparations and competencies are not enough to perform different roles effectively. The presence of skill gaps is mainly attributed to the lack of experience, lack of training and the rapid technological change. On developers' perspective, it is quite challenging because they are exposed to new and frequent updates in technologies (i.e. platforms, programming tools, development frameworks). P15 specified, *"The stuff that is doing in the industry is very far from what is being taught in college."* P17 however argued that, *"The gap is huge simply because the industry moves very fast that's is why the curriculum that has to be taught is outdated."* The lack of experience had been a challenge. Some freshly joined after college and others had gained development experience from other companies but they lack experience in handling the business. On the other hand, the absence of formal training in startups is a concern. Unlike in the corporate organizations, startups did not provide formal training to their employees. Consequently, the available skills of technical people are far from what have been doing or should be doing in the startup industry. They need to know not only the technical aspect of product development but also various facets of business and management. Therefore, failure to address this concern will have a huge impact on the accomplishments of personal and organizational goals.

The Cyclical Basic Social Process of Roling

In this study, Roling indicates a core category that resolves the main concern of developers in technology startups. Roling indicates the act of doing role. The term role is taken from the data and we framed the concept in the form of a gerund, by using an *-ing*. Roling involves two stages: Instalearning and Perfortraying (see Figure 1). The process of Roling holds true for participants holding the assigned role as software developer (i.e. front-end, back-end and full-stacked) as well as those developers occupying the position of founder, chief technology officer, and project manager.

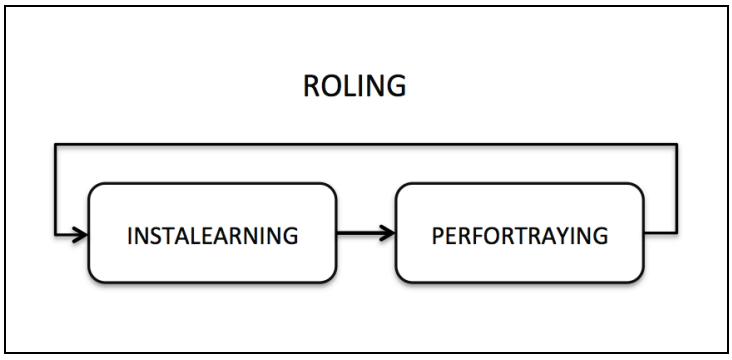

Figure 1. The Basic Social Process of Roling

Instalearning

The first stage in Roling is Instalearning. From the word "instant" and "learning", Instalearning means the rapid way of acquiring skills and knowledge. Instalearning is basic to all developers, whether a neophyte or with previous work experience and those with typical developer role or with founder and manager role, because the work in technology startups is very dependent on the rapid technological updates and frequent market demands. Instalearning in the substantive area can be gained through

self-learning, collaborative learning, and networking. Its core activities, corresponding sub-activities and properties are shown on Figure 2.

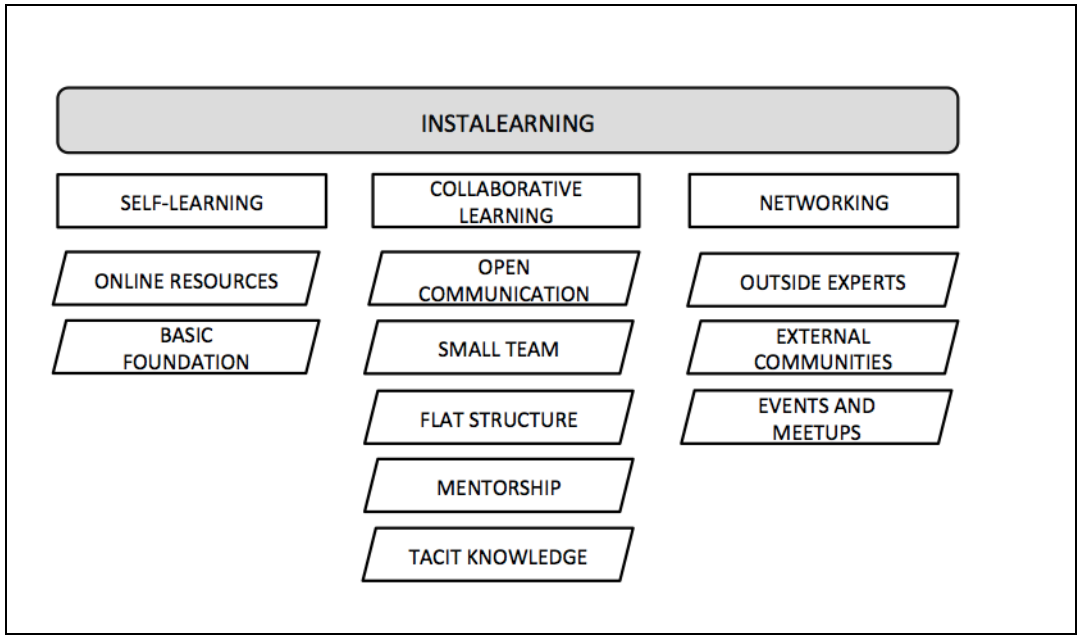

Figure 2. The Instalearning Process of Developers in Technology Startups

Self-learning describes the ability of developers to learn on their own with the aid of reference materials and online resources coupled with strong basic foundation and the determination to learn independently. In resolving the limited skills possessed by developers as well as the rapid updates in technologies and markets, developers simply do self-learning. They first tackled the problems on their own by researching. They regularly read online documentations, tutorials and forums. P5 shared, *"Every day you see people here reading something on the Internet. Technologies moved fast. Even I, I had a hard time to catch up especially on my age now."* P1 also shared his insights, *"I think is very easy because we can always look for resources and tutorials on the Internet."* Thus, the availability of online resources facilitates the learning process. In addition, even when the subject is not on their area of expertise, developers forced themselves to do research because it was needed in the development of their products. They read accounting books for the accounting system, psychology materials for their child development app, legal processes for legal app, among others.

In addition, researching is usually coupled with strong basic foundation and experience, specifically on programming and problem solving. P9 survived the challenges, as he commented, "*It is not hard for me to learn because I am used to work independently. We were not spoonfeeded in college. I think it is with my foundation."* P14 shared a similar experience, "*Even though we have different languages 4 years ago, it is easy for me to learn because of my foundation. What is important is we know the logic."* P2 further emphasized that his background really helped him a lot, "*I was a Philippine Math Olympiad then. Of course, Math is more of a problem solving."* Hence, the aforementioned incidents highlight that the main competencies needed in order to effectively perform developer's role in startups are Math and Logic.

When the resolution of the problem is beyond one's capability, individuals in startups usually collaborate with peers to acquire knowledge. The simple manifestation of learning is by asking help and offering help. P9 explains, *"If I need some help I asked my peers if they do have the knowledge or had encountered the same problem."* On the other hand, by simply saying to peers, *"This is the way to do it",* signifies the importance of sharing knowledge to someone in need. This reciprocal way of knowledge acquisition and sharing occurs naturally within technology startups.

In addition, it should be noted that developers particularly young ones are looking for experienced developers to obtain specific skills. They often consider these experts as their mentors who can advise and help them in solving technical problems. P12 shared, *"If I have concerns, I usually ask <Teammate1> for design and <Teammate2> for programming"*. Developers can tap founders anytime if they need their insights and vice versa.

The abovementioned learning practices occur naturally because power and authority are not implied within individuals. Because of the narrow hierarchical boundaries and having small teams, communication tends to be more open thus enabling the flow of ideas smoothly. It is also observed that discussions and meetings are more informal. P5 shared, *"Sometimes we have a meeting even though we are only two. Sometimes we have a meeting along the pantry."* P10 also added, *"We have no formal process but I think people will just say hey guys, I found out that this is cool thing. What if we do this way? We have that culture where everyone talks to each other".* The social interaction of individuals in an organization facilitates the transfer of knowledge. In particular, the sharing of tacit knowledge among teams leverage innovation and creativity. Aside from physical direct physical interaction, the use of various communication and collaboration tools (i.e. chat, SLACK) facilitates the exchange of ideas.

Developers in technology startups saw the need to interact with other people outside the company to further their knowledge because their personal knowledge and team capabilities were not enough. Accordingly, they created networks and linkages to meet people who can give insights and expertise not only in the development of products but also in managing people and running the business. Most of the respondents who became part of this study were part of the startup communities and actively involved in the developers groups. They attended meetups and conventions to connect with their fellow founders and developers. On our part as researchers, we also participated in various startups' and developers' events to have an idea what is going on their fields. In immersing with them, we observed that they are searching for knowledge. For developers' role, they are yearning to acquire technical knowledge particularly on the latest trends and techniques in developing the product. On the other hand, for developers-founders, aside from the technical aspects of product development they also need insights on the business side, since their main goal is to find a sustainable growth and a scalable business model. P10 shared, *"I tried to talk to fellow founders outside the company about problems with monetization, in dealing with client and many others. It's been very helpful."* In addition, P19 underscored that the psychology aspect in startup, particularly a problem on social isolation should be part of learning and be tackled in the academe. He further added, *"…because in startups you have to handle many works. You need to network so that you can seek help from other people in solving your problem."*

In addition, various incubation hubs across the country provide avenues for these technology startups to grow. Most of the respondents from this study were part of the incubation programs supported by the government, academe, and industry partners. Startups that became part of these programs were not only seeking for physical space but also yearning for learning strategies on how to make their startups successful. Some incubatees were able to learn techniques they provided, however most of the incubation hubs in the country are not yet mature to provide these experts services to their incubatees.

Perfortraying

Perfortraying is the next stage in the basic social process of Roling. Perfortraying is an amalgamation of two words "performing" and "portraying". Performing means *"carrying out"* roles, based on the designated duties and assigned responsibilities. On the other hand, portraying roles means *"depicting"* roles. It indicates that developers are taking other's work temporarily. Developers in technology startups employ several working practices in Perfortraying roles. These practices include: Dutying, Spanning, Substituting, Sharing, Swapping, Grabbing, and Hopping (see Figure 3).

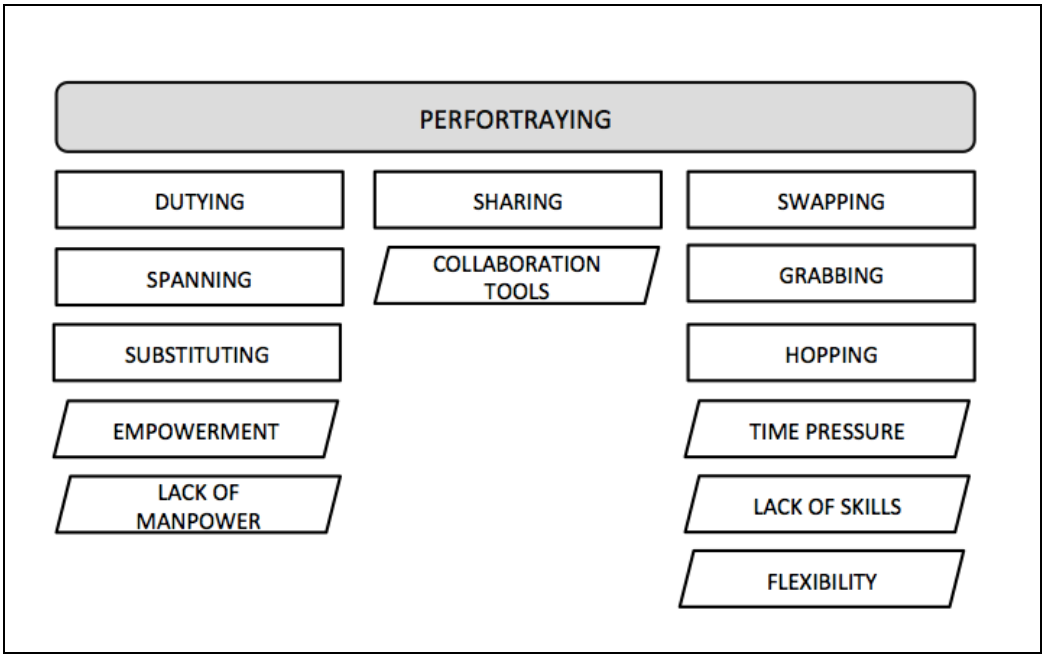

Figure 3. The Perfortraying Process of Developers in Technology Startups

Dutying is performing work based on the designated position. The primary role of a developer is performing the technical works such as front-end and back-end. However, in the startup context some are considered as full-stack engineers who know all the layers in software development. Some specifically assigned to perform development role based on products: app or game. Others specifically in charge to the work based on specific platform: mobile (Android and/or iOS) or web.

Spanning is performing additional roles ranging beyond the assigned roles. In most cases, the work of software developers in technology startups spanned across systems analysis, testing, quality assurance, maintenance, graphics, interfaces, among others. When asked if there is specific person that checks the quality of the product, P11 responded, *"No. I also handle the QA."* Further he said, "*I am a programmer, software developer / engineer. I currently handle the Android version of the app. I do most of the code, web interfaces, and maintenance. I handle the releases at Play store."* For developers who also lead and manage the project, they handled other roles such as marketing and mediating with the clients. For example, P14 specifically emphasized his role as, *"Not only dev. I do take responsibility to market at the same time observing the traction and everything. That's why the founders seen my potential to be a Project Manager."*

For developers-founders, their roles extended beyond technical development of the product. As business owners their works are beyond establishing the company and setting plans. It also includes business, finance, sales, marketing, human resource, legal aspects, and overall aspects of the organization. Thus, they often wear multiple hats especially in the early stage of startups. They didn't restrict themselves in performing defined roles but doing plus other functions including manual tasks and odd jobs. P10 specified, "*Aside from being a founder, I am also a developer, designer, and project manager. I even do fieldworks. I went out to major commute hubs just to interview drivers and operators to know their schedules and how long will it usually take to stop and get off, and when are the rush hours. We all did that just to have data for the commuters' app we are developing. "*

Another practice of portraying role is Substituting, which means acting a role in behalf or in absence of someone. Data revealed that the work of software developers in technology startups is not exclusive to a certain specialized field. Developers can work or requested to work beyond their field of specialization because of the lack of manpower. It happens in many cases for back-end developer doing the job of a front-end and vice versa. P16 shared, "*I am working on the server side programming but there are times that I touched the front-end development because my teammate who*

was assigned to handle the job resigned." This work practice can also be a training ground for some to know both sides and be called full-stacked developer.

Sharing is performing a collaborative work by giving a portion of the work to someone. In most cases, development works in startups are shared or divided among teams noting that the success of the project depends on the collective efforts of teams. P6 recalls, *"The three of us works together in doing the project. I was the one who developed it and mediate with client, <Teammate 1> is the one who design the UX. I gave to <Teammate 2> whatever job assigned."* Sharing development roles depends upon how the task is being divided. It can be based on platform, feature, or module. For instance, P23 shared, *"I am part of the team. We divided features for every task to work. We all work in both mobile and web but we work by feature. When my teammate, who is also a developer, divided the task, she showed us different screens, representing different features. Then, we assigned different person for every feature."* In order to facilitate the accomplishments of individual tasks, teams employ the use of collaboration tools (i.e. SLACK, GitHub). In particular, the use of these tools enabled teams to communicate, track bugs, access and maintain codes, and manage the entire project workflows.

Another practice in performing work is through swapping. Swapping is giving one thing and receiving another thing in return. Exchanging is very common for pairs working on the same projects but different platforms. Since both pairs specialized on something they swap the job. P1 recalls, **"***There are times that we worked for a certain projects, for Android and iOS. My peer worked with one app for iOS. I developed the second app for Android. After we were done, we swap. I developed her project using Android and she developed mine using iOS. We have our own specialization."* Developing a product in a platform not on their specialized fields entails a lot of learning. Thus, the swapping strategy facilitates the completion of projects faster. In addition, the quality of the outputs has not been sacrificed since tasks were handled based on specialization.

Grabbing is a practice of intentionally taking the role of another person. It usually happens due the inability of assigned person to effectively perform the job, intense time pressure, and the absence of formal rules in the organization. P15 emphasized, "*I am a developer but there are times that I was doing the job of a graphic artist because the one that the creative team did was not suited for the app that I was developing. Also there is no fix system in our company.* Though the action caused conflict, the respondent's intent of grabbing the role was not personal but in order to ensure that the product is in good quality. Time element is crucial for technology startups, thus grabbing other's role is a strategy in order for the development to move fast. For example, even graphics designing was not his specialty; P12 took the role because of time pressure. He further explained, "*I am doing all the development jobs for one project. For designs usually it is assigned to Teammate 1 (also the founder) and Teammate 2. But sometimes, when they are too slow I designed up the things. It won't move forward if I don't do anything*." This practice can be a better option for some to accomplish the work early. But this strategy can cause conflict within teams specifically to those who owned the roles. As P6 shared, *"While we set aside the app we were developing, one time our founder was cramming and made some modifications on the features we made. I mean, a founder and senior guy, who supposedly doing the project management and business stuff, still has an urge to develop. I believe he should stop."* From the aforementioned instances, we have also seen that grabbing did not only occur between employees of the same levels, but can also happen between employees and founders and vice versa. Thus, these complexity and flexibility describe the blurring lines of power and authority as well as the execution of roles in technology startups.

Another concept that emerged from the data is Hopping, or moving from one project or teams to another. In the context of technology startups, the team structure is more fluid wherein roles and responsibilities can be shifted and diffused instantaneously. In common cases, many works not only in one team but are part of several teams. P6 enumerated his role, *"Most of the time I am hopping. I was assigned to develop an app name Product A. I was working remotely on Product B. While doing several products I was pulled out to help the development of Product C."* Moreover, members of the team can be pulled anytime as the need arises especially when there are incoming projects and the current project is at the latter stage or more stable stage of development. Accordingly, the team's integration and disintegration describe the dynamics and flexibility of roles in technology startups.

Theoretical Integration with Existing Literature

After identifying the main concern and generating the core category in the substantive area, we validated the preliminary results with the existing literature. The studies of Ries (2012) and Blank (2013) focused on the process management strategies in their Lean Startup methods. We also found out that the existing theories and models are focused on the engineering practices and software development processes (Giardino et al. 2016, Coleman and O'Connor 2008, Coleman and O'Connor 2007, Baskerville 2003) in startups. On the other hand, the focus of our study is on the basic social process of developers working in the context of technology startups.

By mapping the literature, we found out that the core category or main theme of our research varies. However, the concepts generated from our study complement with the causal factors, sub categories and indicators of the existing theories/models. In particular, Giardino et al. (2016) came up with the core category speed-up development in their Greenfield Startup Model. We linked our results to other categories in their study by looking at the contextual conditions and factors that affect speed-up development such as 'severe lack of resources' and 'team as a catalyst of development'. Thus, the result was not inconsistent in acknowledging that developers have big responsibilities because of the limited human resources. In their study, handling multiple roles is one of the indicators in 'team as a catalyst of development' category. In our study, it is considered as the major concept. In addition, prior study conducted by Coleman and O'Connor (2008) revealed that one of the factors that influenced software development processes in startups is the previous experience of founder and software development manager; thus this is also discussed in our study. In his Lean Startup Methodology, Ries (2011) underscored the importance of team empowerment as critical factor in development. In a similar vein, our study also showed how development teams are empowered in carrying out tasks and making decisions. In our study, empowerment is an enabling factor in the Roling process. As such, developers were given autonomy, freedom, flexibility, and power to effectively perform roles. On the other hand, Baskerville et al. (2003) identified that the lack of experience developing software under the following conditions: desperate rush-to-market and new and unique software environment, affect development. Our study yields the same result but with thorough investigation on how the lack of experience is being addressed by developers.

Conclusion and Recommendation

Software developers' work in technology startups is highly regarded, as they are one of the most important factors for its growth and development. In the midst of challenges and uncertainties affecting the industry, this study looks into the human aspect of the phenomenon. By using a GTM, this study uncovered an empirical evidence of their practices and personal experiences.

By abstracting higher-level concepts and inducing patterns in the data, the results of this study found out that the main concern of software developers in their work is to acquire skills in doing multiple roles. The lack of skills is brought by the limited experience of developers and the lack of training, not only on the technical aspect of product development but also on the business and management side of startups. In most cases, developers were not bounded to perform single development role but also include other technical and non-technical works. This observation holds true for both developer position and those developers who have management role.

The core category found in the data that addresses this concern is the cyclical basic social process of Roling. The Roling process has two stages: Instalearning and Perfortraying. This study highlights that despite the time pressure and lack of training, the motivation to self-learn is seen as an enabling force in learning coupled with strong basic foundation and the availability online resources. In addition, learning can be gained through team collaboration. The informal settings, culture of openness and climate of trust among members of the organization facilitate the articulation and acquisition of tacit knowledge. In addition, the results see their needs to further their knowledge by interacting with other people outside the organization, specifically through networks and communities of practice. The results also revealed several strategies in Perfortraying roles. Developers in the context of startups are not limited to perform assigned roles but portray other roles due to the lack of manpower and intense

time pressure. While the flexibility, empowerment and collaboration enables of Roling, the lack of formal structure and rules sometimes cause conflict.

This preliminary result also highlights the needed competencies one must possessed in order to survive in technology startup industry. The results can give substantive evidence to the academe on what is really happening in the substantive area in order for them to strengthen their curricular programs, thereby instilling to the students the preparation and competencies needed in this industry. This study may serve as a reference to the government, private industries, investors, and other stakeholders in advancing the needed support that is critical to both software developers and the technology startup sector.

Lastly, this paper contributes a richer methodological understanding on the use of GTM in information systems field. GTM's simultaneous data collection and analysis through theoretical sampling and constant comparison has contributed greater rigor in studying the underlying phenomenon.

Acknowledgements

The authors would like to thank the participants for sharing their experiences and extending their valuable time during the conduct of this study. Also, a special gratitude to various startup communities and developer groups across the country for mediating with our respondents and for facilitating our requests. Also, we would like to thank the Commission on Higher Education (CHED) through the Faculty Development Program for partially funding the research. Lastly, Ms. Concepcion would like to thank her mother institution, West Visayas State University-Iloilo, for all the support extended.

References

Adolph, S., Krutchen, P., and Hall, W. 2012. "Reconciling Perspectives: A Grounded Theory of How People Manage the Process of Software Development," *Journal of Systems and Software, 85* (6), 1269-1286.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., and Slaughter, S. 2003. "Is" Internet-speed" Software Development Different?," *IEEE Software*, *20*(6), 70-77.

Blank, S. 2007. *The four steps to the epiphany*. K&S Ranch

Blank, S. 2013. "Why The Lean Start-up Changes Everything," *Harvard Business Review* (91:5), pp. 63-72

Blank, S. G., and Dorf, B. 2012. *The Start-up Owner's Manual: The Step-by-step Guide for Building a Great Company*, Pescadero, Calif.: K&S Ranch.

Coleman, G., and O'Connor, R. V. 2008. "An Investigation into Software Development Process Formation in Software Start-ups," *Journal of Enterprise Information Management* , (21:6), pp.633-648.

Coleman, G., and O'Connor, R. 2007. "Using Grounded Theory to Understand Software Process Improvement: A Study of Irish Software Product Companies," *Information and Software Technology*, *49*(6), 654-667.

Fernández, W. D., Lehmann, H., and Underwood, A. 2002. "Rigor and Relevance in Studies of IS Innovation: A Grounded Theory Methodology Approach," *ECIS 2002 Proceedings*, 134.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., and Abrahamsson, P. 2016. "Software Development in Startup Companies: The Greenfield Startup Model," *IEEE Transactions on Software Engineering* , (42:6), pp. 585-604.

Glaser, B. 1992. *Emergence vs Forcing: Basics of Grounded Theory Analysis*. Sociology Press.

Glaser, B. 1998. *Doing Grounded Theory: Issues and Discussions.* Ca: Sociology Press.

Glaser, B. 2004. *Remodelling Grounded Theory: Forum of Qualitative Social Research* (Vol. 5).

Glaser, B. and Strauss, A. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL: Aldine Atherton.

Goulding, C. 2002. *Grounded Theory: A Practical Guide for Management, Business and Market Researchers.* SAGE.

Herrmann, B. L., Gauthier, J. F., Holtschke, D., Berman, R., and Marmer, M. 2015. "The Global Startup Ecosystem Ranking 2015," Retrieved December 14, 2017 from *https://startup-ecosystem. compass. co/ser2015/*

Hose, N. 2013. *Philippine Startup Report.*

Martin, P. Y., and Turner, B. A. 1986. "Grounded Theory and Organizational Research," *The Journal of Applied Behavioral Science*, *22*(2), 141-157.

OECD. 2013. *Development Centre Studies Start-up Latin America Promoting Innovation in the Region.* Organization for Economic Cooperation and Development. OECD Development Centre.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., and Abrahamsson, P. 2014. "Software Development in Startup Companies: A Systematic Mapping Study," *Information and Software Technology*, (56:10), pp. 1200-1218.

Ries, E. 2010. *Startup Lessons Learned* . Retrieved January 12, 2017, from http://www.startuplessonslearned.com/2010/06/what-is-startup.html

Ries, E, 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses,* Crown Business.

Ries, E. 2012. *The Lean Startup Methodology* . Retrieved March 4, 2017, from http://theleanstartup. com/principles

Suddaby, R. 2006. "From The Editors: What Grounded Theory Is Not," *Academy of Management Journal*, *49*(4), 633-642

Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S. S., ... and Edison, H. 2016. "Software Startups–A Research Agenda," *e-Informatica Software Engineering Journal* , (10:1), pp 89-123.

Wiens, J., and Jackson, C. 2015. "The Importance of Young Firms for Economic Growth" in *The Kauffman Foundation and The Public Forum Institute*.